Phase 1

Team 24

Team Members	IDs
Abdelrahman Mohamed Ezzat	1190158
Ali Mohamed Ali Hashish	1190223
Farah Mohamed Abdelfatah	1190176
Ziad Ahmed Hamed	4200002

Project Idea:

Face and Image Cartoonization:

We offer the user the choice to either only cartoonize the face or cartoonize the whole frame captured using a video (or web-cam).

Block Diagram:

Overall Input: A video.

Overall Output: The cartoonized video.

Original Image:

The original image before any preprocessing.

This is the input to the upcoming stages.

Cartoonize Face only?:

A parameter for the user to decide whether they want to cartoonize the face only, or the whole image.

Input: A parameter stating whether the user wants to cartoonize only the face or the whole image.

Face Detection:

Detecting the face to determine which part will be cartoonized. We may use LBPH (Local Binary Pattern Histogram).

Input: The original image.

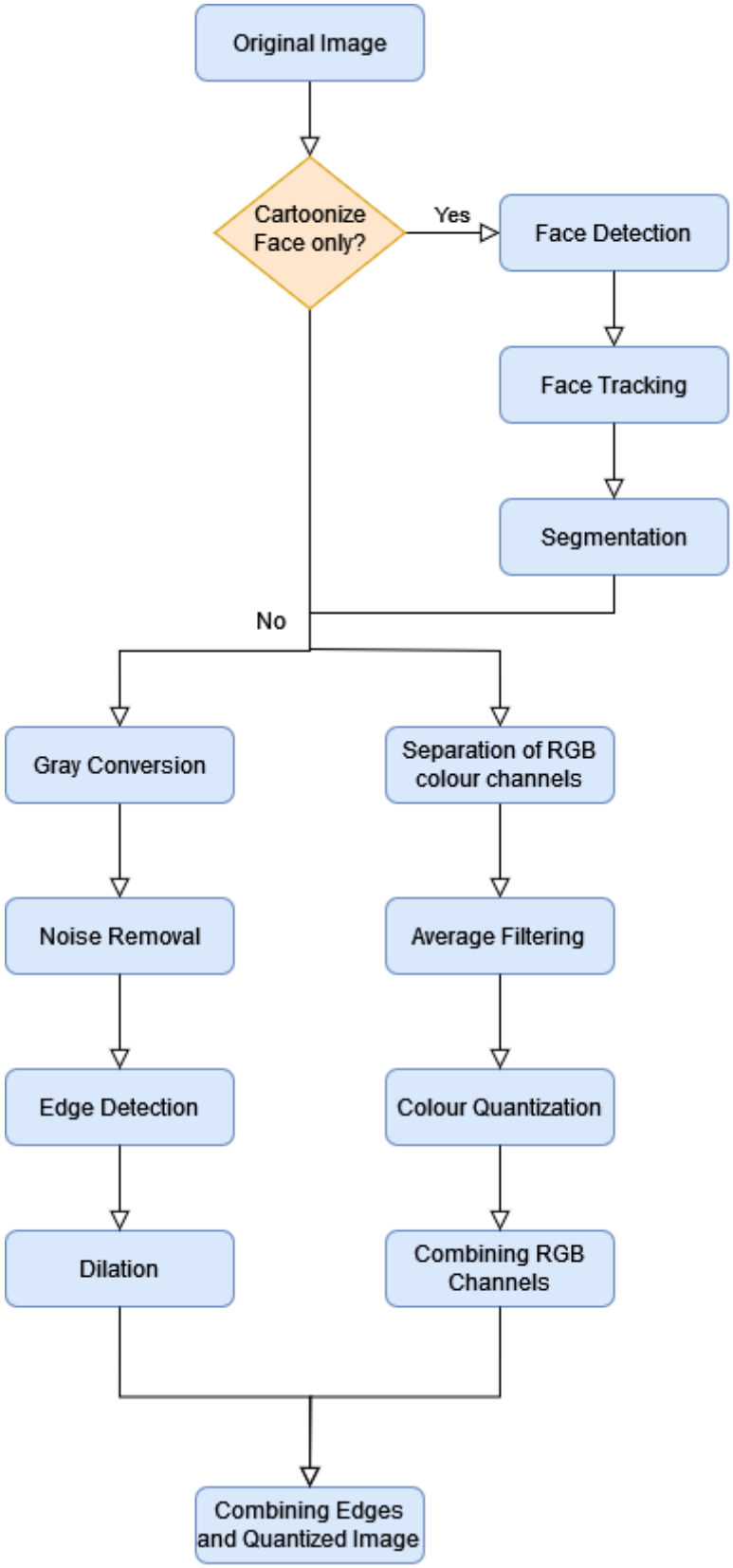
Output: The part containing the face (bounding box).

Face Tracking:

Keeping track of the face, to overlay the cartoonized image. We can use mean-shift or tracking by detection.

Input: History of the part detected from the face (the previous face detection outputs).

Output: The part containing the face being tracked.



Segmentation:

Segment the face, so that the cartoonization is applied to only the face.

Input: A bounding box of the face part.

Output: An image separating the face from the original image.

Gray Conversion:

Converting the image to gray image, to minimize the number of operations needed, improving performance.

Input: Original Image (or the detected face part of the image).

Output: Gray scaled image.

Noise Removal:

Removing unwanted noise from the image, to obtain better results. We can use Median filtering.

Input: Gray scaled image.

Output: A smoothed image without noise.

Edge Detection:

Detecting edges using canny edge detection. This will be used to enhance the lines and edges in the cartoonized image, giving it a more defined look.

Input: Image after noise removal.

Output: Image with edges.

Dilation:

Thickening the edges so that they are more apparent and clearer.

Input: Image with edges.

Output: Image with thicker edges.

Separation of RGB Colour Channels:

This is done so that the upcoming steps are performed on each channel individually, to produce more agreeable results.

Input: Original Image (or the detected face part).

Output: 3 images, one for each channel.

Average Filtering:

Smoothing the 3 colour channels to give a blurring effect. This helps in reducing colour quantities, as each pixel would be affected by its neighbouring ones.

Input: The 3 images.

Output: Blurred images.

Colour Quantization:

Decreasing the number of colours compared to the original image, since a cartoonized colour palette is not as rich as the normal one.

Input: Blurred images.

Output: Quantized colours for each channel (RGB).

Combining RGB Channels:

Combining the 3 channels again to obtain an RGB image.

Input: 3 quantized images

Output: 1 RGB image

Combining Edges and Quantized Image:

Combining the output of both branches, to obtain the final cartoonized image.

Input: Edges and quantized RGB image.

Output: Cartoonized image.

Additional Comments:

We might use Bilateral filter for noise removal instead of Median filtering. In which case this would require a non-primitive function (we will decide after comparing results).

References:

- S. K. Shrivastava and R. Gajjar, "Image Processing based Image to Cartoon Generation: Reducing complexity of large computation arising from Deep Learning," 2023 International Conference on Computational Intelligence and Sustainable Engineering Solutions (CISES), Greater Noida, India, 2023, pp. 650-656, doi: 10.1109/CISES58720.2023.10183524.
- Dandan Che and Zhenjiang Miao, "Real-time cartoon style video generation," 2010 International Conference on Image Analysis and Signal Processing, Zhejiang, China, 2010, pp. 640-643, doi: 10.1109/IASP.2010.5476191.